Noname manuscript No.
(will be inserted by the editor)

Volcano Eruption Algorithm for Solving Optimization Problems

Eghbal Hosseini · Ali Safaa Sadiq · Kayhan Zrar Ghafoor · Danda B. Rawat · Mehrdad Saif · Xinan Yang

Received: date / Accepted: date

Abstract Meta-heuristic algorithms have been proposed to solve several optimization problems in different research areas due to their unique attractive features. Traditionally, heuristic approaches are designed separately for discrete and continuous problems. This paper leverages the meta-heuristic algorithm for solving NP-hard problems in both continuous and discrete optimization fields, such as nonlinear and multi-level programming problems through extensive simulations of volcano eruption process. In particular, a new optimization solution named Volcano Eruption Algorithm (VEA) proposed in this paper, which is inspired from the nature of volcano eruption. The feasibility and efficiency of the algorithm are evaluated using numerical results obtained through several test problems reported in the state-of-the-art literature. Based on the solutions and number of required iterations, we observed that the proposed meta-heuristic algorithm performs remarkably well to solve NP-hard problem. Furthermore, the proposed algorithm is applied to solve some large-size benchmarking LP and Internet of Vehicles (IoV) problems efficiently.

Eghbal Hosseini
Research Center of Mathematics, Polytechnic University-Erbil, Erbil, Kurdistan E-mail: kseghbalhosseini@gmail.com

Ali Safaa
Wolverhampton Cyber Research Institute, School of Mathematics and Computer Science, University of Wolverhampton, Wolverhampton, WV1 1LY, UK; Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Fortitude Valley, Brisbane, 4006 QLD, Australia. E-mail: ali.sadiq@wlv.ac.uk

Kayhan Zrar Ghafoor
Department of Software Engineering, Salahaddin University-Erbil, Iraq; School of Mathematics and Computer Science, University of Wolverhampton, Wulfruna Street, Wolverhampton, WV1 1LY, UK. E-mail: kayhan@ieee.org

Danda B. Rawat
Department of Electrical Engineering and Computer Science at Howard University, Washington, DC 20059, USA. E-mail: db.rawat@ieee.org

Mehrdad Saif
Department of Electrical and Computer Engineering, University of Windsor, Windsor, N9B 3P4 Canada. E-mail: msaif@uwindsor.ca

Xinan Yang
Department of Mathematics , University of Essex. E-mail: lxyangk@essex.ac.uk

Keywords Optimization · meta-heuristics · Constrained Optimization · Volcano Eruption Algorithm (VEA) · Bi-level Optimization

1 Introduction

We have witnessed fast developments in the field of nature-inspired algorithms in the past few years. The popularity of nature-inspired algorithms have been possible as a result of their promising applications in solving engineering problems. In particular, these algorithms enable gradient-free mechanism and avoid local optima. The first advantage of meta-heuristic is that it does not require the derivative of the search space and leads to finding many good solutions. Properties of guided random search technique as well as exploration and exploitation make meta-heuristic algorithms to avoid getting trapped in a local optima. As a result, there are several applications of such algorithm in many engineering applications [1].

Meta-heuristic algorithms can be used to train neural network in solving real-life problems, though every approach has its own limitations. Some of the prominent meta-heuristic algorithms include Particle Swarm Optimization (PSO) [1] and Autonomous Particles Groups for PSO [22] , (AGPSO) [23] Bat Algorithm (BA) [7] and its recent application in optimizing beamforming for mmWave in 5G communication [24], Fire Fly (FF) [10]. On the other hand, nature creature inspired algorithms have also been proposed to solve optimization problems, such as: Whale Optimization Algorithm (WOA) [25], Ions Motion Optimization (IMO) [26] and Grey wolf optimizer (GWO) [27]. While together with those inspired by nature phenomena, such as: Chaotic Gravitational Search Algorithm (CGSA) [28] and the recent application of Multi-Verse algorithm in optimizing the accuracy of fraud detections in smart e-commerce ecosystem [29]. However, no heuristic algorithm is the best suited to solve all optimization problems. Moreover, the limitations of high computational cost and premature convergence, the difficulties of selecting best tunable parameters such as the mutation/corssover rate, cut-off time etc. all raise the needs of designing more advanced approaches.

In machine learning, classification in a supervised learning process refers to the process of computer learning to which class of data a new set of observation belongs. This is based on a prior learning conducted on a labelled training dataset. Evolutionary or nature-inspired meta-heuristic algorithms can be a good option in the process of designing/training a classification system. As an example, Support Vector Machine (SVM) is an efficient supervised learning algorithm that can be applied for classification [8]. The optimization of SVM parameters is possible through algorithms like PSO or FF. Feature selection plays a vital role in the process of classification. It turns out that feature selection can be achieved through parameter optimization of SVM using a meta-heuristic algorithm [8]. Feature selection through this process is another example of an application area where a meta-heuristic approach could be effective. It should be noted however that there are certain challenges with SVM such as: its high algorithmic complexity which leads to higher computational cost, extensive memory requirements, and selection of appropriate kernel parameters which may be tricky [10].

As a result, success of a meta-heuristic approach in one instance, may not guarantee a similar success in another. Researchers have proposed meta-heuristic approaches designed for solving specific problems (e.g. see [1-3] and references cited therein). They have tried to solve optimization problems by simulating several algorithms based on behavior of animals and insects, natural phenomena, or scientific theories [4-14]. Some of these proposed algorithms are: artificial bee colony algorithm [5], krill herd algorithm [6], social spider optimization [8], chicken swarm optimization (CSO) [9], big bang algorithm (BBA) [11], laying chicken algorithm (LCA) [12,19], modified genetic algorithm [13], [30], combined meta-heuristic and classic algorithm [14]. Almost all previous meta-heuristics have been inspired from behavior of animals or insects and only one of them has been simulated from a scientific theory [11].

This paper proposes, for the first time, an algorithm which is inspired from a natural event, that is, volcano eruption. The algorithm is a novel optimizer for solving various types of continuous and discrete optimization problems. The main contribution of this paper is the translation of natural process of volcano eruption that formulated our proposed Volcano Eruption Algorithm (VEA) to be used as an optimizer. VEA optimizer has gained its robustness from the nature concept of volcano in generating the initial population, movement of solutions, explosion, and eruption in the space. Furthermore, the proposed VEA optimizer could achieve an acceptable computational complexity in comparison with the state of the art. The reason was that the proposed algorithm originates from a scientific process, involves simple steps, and implementation. However, the algorithm requires a high number of solutions in some iteration as its inherent behaviour in changing all feasible solutions in different directions. Though, VEA provides acceptable best solution in comparison to other meta-heuristics algorithms. This is because it uses different exploration directions, (due to explosions and eruptions), and large region of feasible space. Eventually, our proposed VEA could contribute in solving wide range of linear, non-linear, multi-level, multi-objective, and transportation based on IoV complex optimization problems.

The rest of this paper is organized as follows. Section 2 provide the motivation behind the proposed VEA. Section 3 provide the literature review on recent advances of developed meta-heuristic algorithms. This is followed by presenting an overview of the proposed approach and details of the designed algorithm in section 4. Section 5 presents the experiments and computational results which are conducted in the paper. Finally, Section 6 concludes the paper.

2 Inspiration

The nature of volcano has motivated the development of our new optimization algorithm that called VEA. VEA optimizer mimics volcano eruption, which is an opening or a hole on the earth's surface that acts as a vent for release of pressurized gases, ashes, and molten rock or magma deep beneath the surface of earth. Deep underground, pressurized magma is passed through a passageway or a conduit, called the volcanic pipe. Magma is referred to as lava when it reaches the hole on the surface of earth and erupts out of it [15]. There are a number of stages leading to formation of a volcano that can be summarized as follows:

1. Rise of magma through cracks in the earth.
2. Build up of pressure.
3. Volcanic eruption and rise of magma to earth's surface.
4. Formation of a crust as a result of lava's cooling down.
5. Repetition of this process over time leading to several layers of rock that builds up over time resulting in a volcano.

Taking into account the aforementioned volcano eruption's stages, a new meta-heuristic VEA algorithm is introduced. In the process of volcano eruption, mass of magma is needed at the first step of this process, so VEA starts with some solutions as initial population. In the volcano eruption process, magma rises through pipes; hence, similar idea is used to move some of solutions in different directions for certain determined distances. In the next step, all solutions will come down and move again in different directions just like the process of volcano eruption.

Finally, some of the solutions in the *"pipes"*, and points near the surface of earth, are exploded in the region of optimization programming problem. This step comes from eruption of volcano at the top of the mountain into the space. Best solution of all populations will be found, and the algorithm will be using it as an initial solution for the next iteration. As VEA optimizer progresses, it changes and modifies the population and set of solutions, in each iteration. To sum-up, the movement of magma from inside the ground to top of mountain and its explosion have motivated in formulating the main concept for simulation of our proposed VEA optimizer.

3 Related Work

There are two main classes of optimization algorithms. The first class known as deterministic while the second named stochastic method. When an optimization algorithm that works over a deterministic method, it could be whichever gradient-based or non-gradient based type. The gradient-based method that deployed to locate global solution is a method where mathematical programming is used. Gradient method is incorporating linear and non-linear programming [35]. In contrast, using a condition-based method, another type of optimization algorithms could be formulated to find the global solution of a given problem [36]. One of the main issues facing mathematical programing approaches is that the trapping within the local optima solutions while searching for a feasible solution in a non-linear problem.

Hence, many research studies have recently been carried out as a way to overcome this issue by developing some of the existing optimization approaches or hybrid them with different types of algorithms. In some instances, an optimization algorithm has been developed to uniquely address a specified problem, while makes it limited and not generalized to a wide-range of optimization problems [37]. The other challenge that could be experienced while developing a non-gradient (deterministic method), their implementation required a sophisticated mathematical modeling [38]. Therefore, the use of meta-heuristic algorithms has emerged to overcome such kind of challenges, as they are much easier to understand and adopt. Though, such kind of algorithms is classified as a stochastic optimization that requires random operators. These operators and other random variables will help meta-heuristics during their global search and avoiding them from trapping into a local solution of a given problem.

Meta-heuristic algorithms are inspired from either the behavior of animals, insects, or certain natural events. Chemical pheromone of ants is the fundamental concept used for ant colony optimization [39 and 40]. While the direction and global best have inspired the foundation of particle swarm optimization (PSO) [1]. In contrast, fire fly algorithm [41] has simulated the light indication of fireflies. Similarly, laying chicken algorithm [12] is simulated based on warming of eggs, (heat distribution between eggs), as the main concept in formulating the exploration and exploitation strategies.

On the other hand, the authors in [42] have proposed a novel optimization method that inspired from one of the theories of the evolution of the universe, called the Big Bang and Big Crunch (BB–BC). In the BB-BC, two phases are formulated. At the first phase of BB, random points are generated. While in the second phase of BC, these generated points are shrinked to a single demonstrative point. This was achieved using an approach called a center of mass or minimal cost. The authors in [31] have examined the exploration-exploitation strategies related to multi-armed bandit settings. They have introduced an adaptive clustering technique for content recommendation. The algorithm considers the collaborative effects that occur due to the interaction of the users with the items.

As another attempt, the authors in [32] have introduced a distributed clustering for solving linear bandits in peer-to-peer networks with the presents of controlled communication facilities. On the other hand, mining λ-Maximal Cliques from a Fuzzy Graph was introduced in [34], and the Stochastic Optimization Techniques are deployed for quantification performance measures by the authors in [35]. In spite of all the aforementioned related work, we could observe that different natures of problems require various optimization methods to support efficient and cost-effective approaches. To this end, in this paper we have presented a new optimization algorithm that inspired from the nature of volcano to formulate Volcano Eruption Algorithm (VEA), which is detailed out in the next section.

4 The Proposed Volcano Eruption Algorithm (VEA)

In this section, we present the details of the proposed VEA. More particularly, we discussed mathematical equations, details of VEA simulation, and various steps to find the optimal solution in several types of optimization problem.

4.1 The Solutions and Populations of VEA

VEA starts with initial solution that is created randomly and initial population is generated as magma in the volcano eruption process. In fact, initial population in VEA represents the mass of magma below the surface of earth. In volcano eruption process, after the formation of magma, it is distributed in different directions through pipes (points near surface of the earth) and rising toward the surface of earth. Similar to this natural phenomenon, the solution of initial population is distributed in different directions. In the initial population, each possible solution x_i is created randomly in proximity to the initial solution x_0. To form possible solutions, one of following probability distribution functions are used: a) Probability function of the binomial distribution; b) probability function of the geometric distribution; c) probability function of the hypergeometric distribution; d) probability function of the Poisson distribution, and according to the following formula:

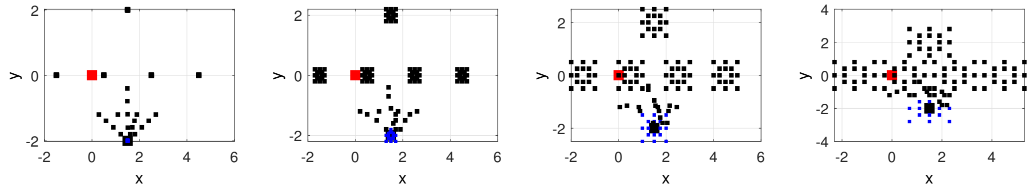
$$||x_i - x_0|| \leq \epsilon \tag{1}$$

where in R^n, i=1,2,...,n , ϵ is a small positive number.

Fig. 1 shows the movement of initial population and generation of the possible next population by varying the value of ϵ from 0.01 to 0.4. We can observe feasible solutions in the initial population (small blue points) and their movement in different directions for a given problem. More importantly, the point with red color in the figure is the optimal solution. Further, in Fig. 1, the next population is represented as black points and distributed in random directions based on the following equation:

$$x_{j+1} = x_j + \lambda * d_{rj} \tag{2}$$

where d_{rj} is the j^{th} random direction to reach the solution.



(a) Epsilon=0.01 (b) Epsilon=0.1 (c) Epsilon=0.25 (d) Epsilon=0.4

Fig. 1 Movement of initial population and generation of the next population by different values of ϵ.

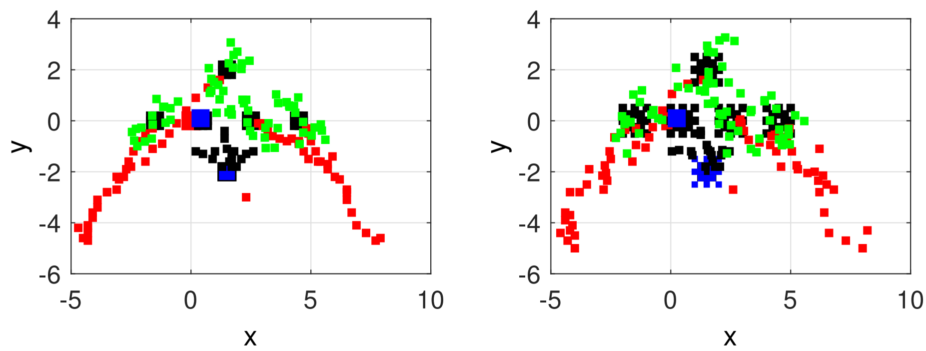
Algorithm 1 shows the procedure of generating initial solution and population. The psudo code starts with a set of random initial feasible solutions and then generating initial population near initial solution according to the formula 1. Thereafter, at each iteration, the algorithm generates a solution of population based on formula 2.

4.2 Explosion and Eruption of VEA

In this section, we present the solutions of the current population represented as black points in Fig. 2. These black points are then exploded (represented as green points) and erupted (shown as red points) in feasible search space. This mimics the explosion and eruption of volcano at the top of the mountain. In fact, solutions are changed in direction of the vector,

Algorithm 1 Pseudo code of solutions and populations

```
n: Number of solutions
ϵ: A given small positive number
Generate a random initial feasible solution
Generate initial population near initial solution according to the formula 1
for i=1 to n do
    Generate solution of population based on formula 2
end for
```



(a) Epsilon=0.1 (b) Epsilon=0.25

Fig. 2 Eruption and explosion of the population by considering different values of ϵ.

which connects solutions and the center solution of the population. These movements are according to the equations 3 and 4:

$$x_{j+1} = x_j + \alpha d_{cj} \tag{3}$$

$$x_{j+1} = x_j - \beta d_{cj} \tag{4}$$

where d_{cj} is the vector to connect x_c, x_j, α *and* β. It is worth mentioning that α *and* β are positive constants and x_c is the solution derived from initial solution in the previous direction to compose a population (center solution). Moreover, equation 3 represents the formulation of the explosion phenomenon, and equation 4 represents the eruption process.

In the process of explosion and eruption phenomenon, the proposed VEA finds the best solutions in all populations. The populations include: initial population, second population which is created after movement of initial population in different directions (black points), third population which is generated after explosion using equation 3 (green points), and finally the fourth population which is constructed after eruption and using equation 4 (red points). Then, the best solution can be found and is shown by large blue point in Fig. 2. The VEA continuously search for the optimal point using the best solution as initial solution for the next iteration. The process of explosion and eruption in the first iteration has shown in Fig. 2 with ϵ=0.1, ϵ=0.25. Algorithm 2 shows the pseudo code for explosion and eruption.

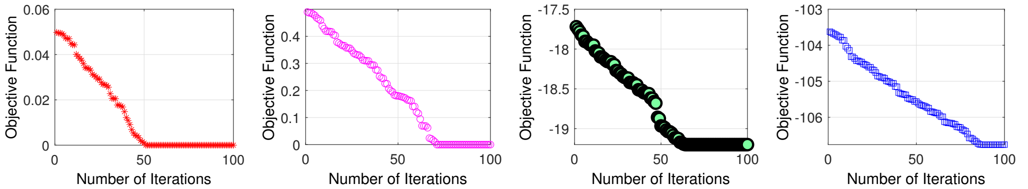
4.3 Analysis of VEA Convergence Behavior

In the previous section, we have shown that VEA intelligently explores the promising regions of the search space and targets the best one. The VEA promptly replaces initial solutions with the best ones and then progressively converge. To achieve this goal, The procedure of the proposed algorithm is summarized as follows:

1. Initial solution is generated randomly. It will be the origin for constrained problems.
2. Initial population is generated near to the initial solution. Here, ϵ is a given positive small number and j=1. The VEA finishes the search process when meets the termination condition.
3. Solutions are moved into different directions for a specific distance. (Until reach the pipes)
4. New solutions near pipes are generated.
5. Explosion of the solutions are performed near pipes.
6. Falling of solutions near pipes from different directions.

Algorithm 2 Pseudo code of explosion and eruption

```
λ is given constant
n is the number of initial population
for j=1 to n do
    x_{j+1} = x_j + λd
end for
for i=1 to n do
    x_{i+1} = x_i + λd
end for
```



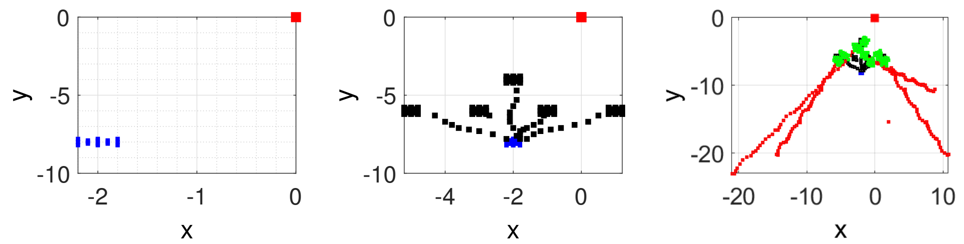
(a) The process of convergence example 4 (b) The process of convergence example 5 (c) The process of convergence example 6 (d) The process of convergence example 7

Fig. 3 The process of finding optimal solution (convergence) by VEA- Examples 4-7.

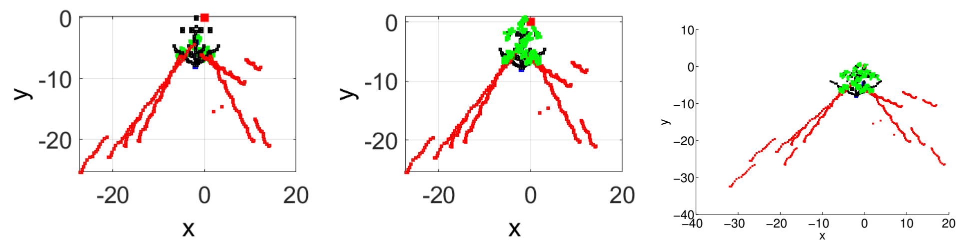
7. Find the best solution of the population. If $j < 2$, (let j=j+1) go to the step 2 with the best solution serving as the initial solution to the next population. For instance, Fig. 3 shows the process of convergence for examples 4 to 7.
8. VEA is terminated by reaching the termination condition $d(f(x_{j+1}), f(x_j)) < \epsilon$ and converges x_{j+1} as the best solution whereas x_j is the best solution in the j^{th} iteration. If the termination criteria is not satisfied, set the value of j to j+1 and go to step 2. In Fig. 4, the aforementioned steps and the progress of the algorithm to find optimal solution R^2 are illustrated. Furthermore, we defined d in equation 5:

$$\max_i \quad |f(x^i_{j+1}) - f(x^i_j)| = d(f(x_{j+1}), f(x_j)) \tag{5}$$

Convergence behavior and property of any meta-heuristic algorithm are very significant. To achieve this goal, the following theorem proves that the proposed VEA algorithm is convergent.



(a) Steps1,2 Initial Solutions and OS (b) Steps 3,4 Solutions Near Pipes (c) Step 5 Eruption and Explosion



(d) Step 6 Best Solution,New Population (e) Iteration 2 Explosion of Population (f) Iteration 2 Optimal Solution

Fig. 4 Steps of the algorithm to solve a given problem.

Theorem 1 The sequence of F_k, which is proposed in the procedure of VEA, is convergent to the optimal solution. Note that F_k is defined as an objective function at point x(k).

Proof Let $(F_v) = (F(t^v)) = (F(t_1^v), F(t_2^v), ..., F(t_n^v)) = (F_1^{(v)}, F_2^{(v)}, ..., F_n^{(v)})$
According to step 6 in the procedure of the proposed algorithm (section 4.3)

$$\max |f(x_{j+1}^i) - f(x_j^i)| = d(f(x_{j+1}), f(x_j)) = d(F_{j+1}, F_j) < \epsilon_1$$

Therefore $|f(x_{j+1}^i) - f(x_j^i)|$ for each i. There is large number such as N which $k+1 > k > N$ and $j = 1, 2, ..., n.$. Now we have:

$$|F_j^{(k+1)}, F_j^{(k)}| < \epsilon_1$$

Now let m=k+1, r=k then we have:

$$|F_j^{(m)}, F_j^{(r)}| < \epsilon_1 \ \ For \ \ m > r > n$$

This shows that for each fixed j, $1 \leq j \leq n$, the sequence $(F_j^{(1)}, F_j^{(2)}, ...)$ is Cauchy of real numbers, then it converges, say to F_k. Using these n times, we define $(F_1, F_2, ..., F_n)$ and if m=k+1, r=k,

$$d(F_m, F_r) < \epsilon_1$$

Now if F_k we have

$$d(F_m, F_r) \leq \epsilon_1$$

This shows that F is the limit of F_m and the sequence is convergent. Thus, this is considered a proof of the Theorem.

4.4 Mathematical Nature of the Algorithm

This section presents the mathematical background of the VEA and summarized in the following points:

1. Generation of feasible initial solution and population.
2. Movement of solutions to improve population and reaching better solutions.
3. Termination of the algorithm when it reaches the best solution.
4. Convergent of the algorithm.

Firstly, a feasible solution is created randomly in the feasible region. So to produce feasible initial population, generated near enough to the initial solution based on the following formula: $\sqrt{(x_{i1} - x_{01})^2 + +(x_{i2} - x_{02})^2 + ... + (x_{in} - x_{0n})^2} \leq \epsilon$ In this phase, the algorithm tries to move solutions of initial population in different random directions to increase the chances for finding better solutions. This movements is based on equation 2. In the second phase, explosion and eruption of volcano is simulated by going up and then coming down based on equations 3 and 4. The third phase satisfies the termination of the algorithm based on the following condition:
If $d(f(x_{j+1}), f(x_j)) = Max|f(x_{j+1}^i) - f(x_j^i)| < \epsilon$, then the algorithm will be finished and x_{j+1} is the best solution by VEA and x_j is the best solution in jth iteration.
Finally, convergence feature of VEA has been proven by the afore-said condition and Theorem 1.

5 Computational Results

To show the numerical efficiency of the proposed VEA, several mathematical optimization problems are addressed and solved using our proposed VEA optimizer. In particular, two classes of optimization problems are considered and solved: a) continuous problems with small size, and b) discrete and practical problems with large size. Then, VEA is used to solve complex routing optimization NP hard problem in harsh IoV scenarios.

5.1 VEA Solving Continuous Problems

This section presents almost all kinds of continuous optimization problems: constrained, unconstrained, linear, non-linear, multi-level and multi-objective will be solved by proposed VEA.

Example 1 [16](Constrained - Non-linear)

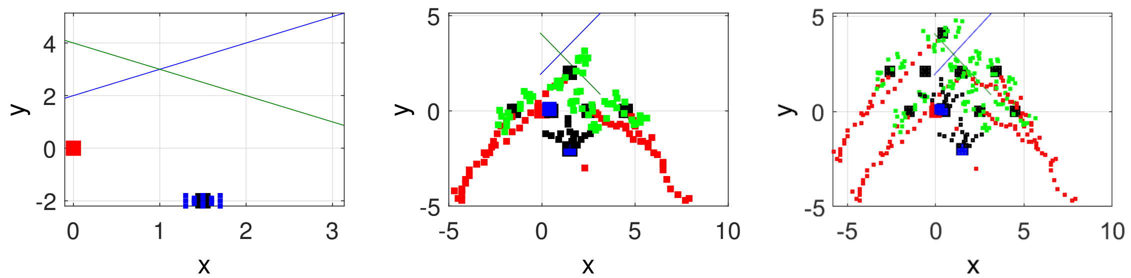
The initial and optimal solutions as well as different populations of the algorithm for Example 1 are shown in Fig. 5. The large blue point in Fig. 5 is the optimal solution, which has been found by solutions after 2 iterations.

In order to compare the proposed VEA with classical methods, we consider the following non-linear programming problem; as shown in equation 6:

$$\begin{aligned} \min \quad & -(x_1-4)^2-(x_2-4)^2 \\ \text{s.t.} \quad & x_1-3 \le 0 \\ & -x_1+x_2-2 \le 0 \\ & x_1+x_2-4 \le 0 \\ & x_1, x_2 \ge 0 \end{aligned} \tag{6}$$

Example 2 [17] (Multi-Level)

Consider the following linear bi-level programming problem:



(a) Solutions near(1.5,-2) and optimal solution (b) Generation 1 (c) Generation 2

Fig. 5 Finding optimal solution by VEA-Example 1.

Table 1 Comparison of VEA with non-linear algorithms- Example 1,2

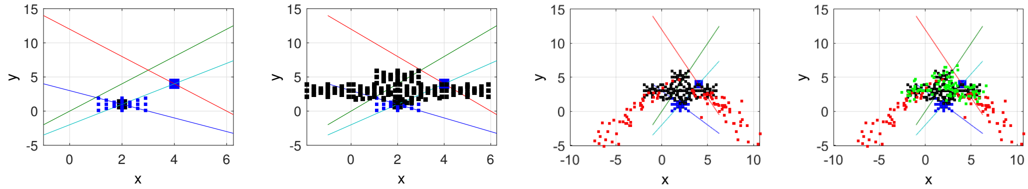
Algorithms	**N. Agents**	**N. Iterations**	**Optimal Solution**	**F Min**	ϵ	**Initial Solution**
Example 1 - VEA	16	1	(0.4,0.09)	-28.17	1	(1.5,-2)
VEA	16	2	(0.2,0.09)	-29.65	1	(1.5,-2)
Exact Method [16]	None	None	(0,0)	-32	None	None
Example 2 -VEA	24	1	(3.4,3.1)	-9	1	(2,1)
VEA	24	2	(4,4)	-12	1	(2,1)
Exact Method [18]	None	None	(4,4)	-12	None	None
Other Methods [17]	None	None	(3.9,4)	-12.1	None	None

$$
\begin{aligned}
&\min x - 4y \\
&subject\ to \\
&\min y \\
&subject\ to \\
&x + y \geq 3 \\
&-2x + y \leq 0 \\
&2x + y \leq 12 \\
&3x - 2y \leq 4 \\
&x, y \geq 0
\end{aligned}
\tag{7}
$$

Using Karush–Kuhn–Tucker (KKT) conditions, the problem will be converted into the following problem:

$$
\begin{aligned}
&\min x - 4y \\
&subject\ to \\
&-\lambda_1 + \lambda_2 + \lambda_3 - 2\lambda_4 = -1 \\
&\lambda_1(-x - y + 3) = 0 \\
&\lambda_2(-2x + y) = 0 \\
&\lambda_3(2x + y - 12) = 0 \\
&\lambda_4(3x - 2y - 4) = 0 \\
&-x - y + 3 \leq 0 \\
&-2x + y \leq 0 \\
&2x + y - 12 \leq 0 \\
&3x - 2y - 4 \leq 0 \\
&x, y, \lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0
\end{aligned}
\tag{8}
$$

The bi-level programming problem is NP-Hard because of its two objective functions. In fact, these two objective functions should be optimized in two different levels at the same time. Therefore, proposing a solution for this problem is significant. The proposed VEA optimizer could find optimal solution in a fast pace (within 2 iterations as shown in Table 1), which is the exact solution that found by algorithms with relatively small number of agents. By solving such problems presented in examples 1 and 2, VEA shows its high performance with less complexity. Behavior of solutions, constraints of the problem and optimal solution have been shown in Fig. 6.



(a) Solutions near (2,1) and optimal solution (b) Generation 1 (c) Generation 2 (d) Generation 2

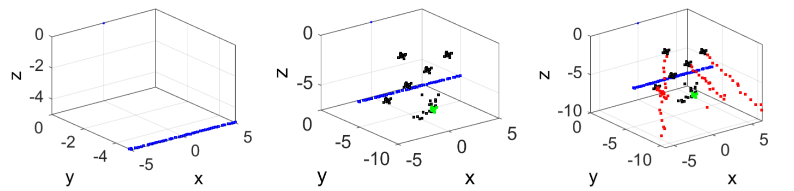
Fig. 6 Process of finding optimal solution by VEA- Example 2.

Example 3 [20] (Multi-Objective)

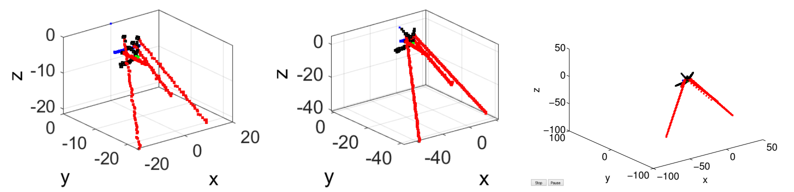
In this example, VEA is used for solving DTLZ benchmark problems [21]. Behavior of the algorithm in finding the pareto optimal for DTLZ1 problem is shown in Fig. 7. It is clear that some of solutions in the population have reached to pareto optimal solution; this illustrates the feasibility of the algorithm as shown in Fig. 7c and 7d, which also indicate the initial population. Moreover, efficiency of the algorithm is obvious by comparing of Fig. 7a and 7f. At the begining of applying the algorithm most of solutions are completely far from pareto optimal. However, during the searching process, the algorithm solutions achieve pareto optimal. Fig. 7f shows that the last population has surrounded pareto optimal solutions.

Table 2 shows the comparison of best solutions to get Pareto optimal of DTLZ problems by VEA and the best method in [21].

Example 4



(a) Solutions near (1,-1) and OS (b) Generation 2 (c) Generation 2 and optimal solution

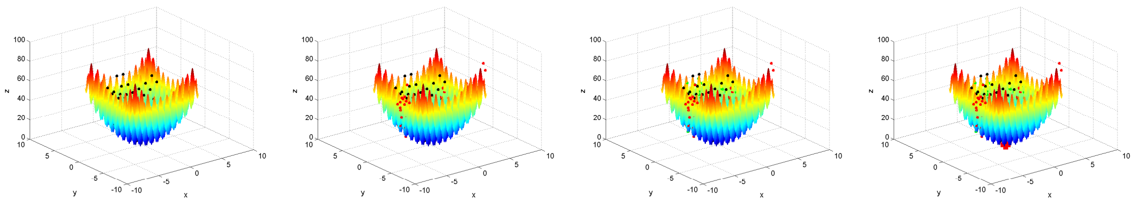


(d) Generation 2 (e) Generation 2 (f) Iteration 2 Optimal Solution

Fig. 7 Generations of VEA to find optimal solution - Example 3.

Table 2 Comparison of VEA and other methods for DTLZ problems

Problems	k	ParEGO			VEA		
		min	mean	max	min	mean	max
DTLZ1	3	13.42	52.47	112.7	9.13	31.24	78.18
DTLZ1	4	18.63	45.45	87.76	11.57	32.21	59.32
DTLZ1	10	NA	NA	NA	1.12	1.78	2.95
DTLZ2	3	0.151	0.191	0.243	0.093	0.105	0.164
DTLZ2	4	0.289	0.337	0.408	0.099	0.187	0.275
DTLZ2	10	NA	NA	NA	0.081	0.123	0.187
DTLZ3	3	81.15	145.5	261.6	52.56	123.26	213.77
DTLZ3	4	66.93	138.1	209.4	43.32	107.41	186.24
DTLZ3	10	NA	NA	NA	0.85	1.14	1.96



(a) Initial population (b) Generation 2 (c) Generation 4 (d) Generation 4 and optimal solution for RF

Fig. 8 Process of finding optimal solution by VEA- Example 4

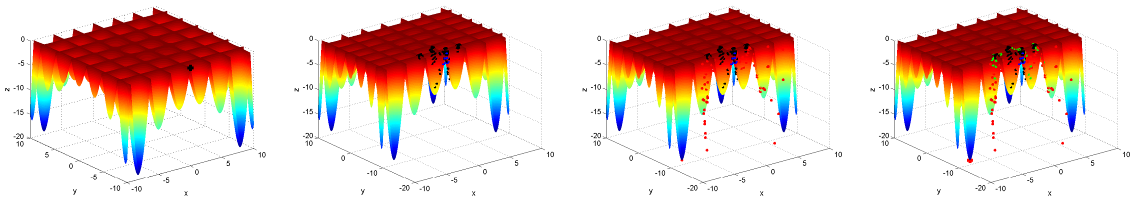
In this example, we apply our proposed VEA on non-convex optimization problem named Rastrigin Function (RF). Fig. 8 shows the process of finding optimal solution using VEA for Example 4.

$$\min 20 + (x^2 - 10cos(2\pi x)) + (y^2 - 10cos(2\pi y)) \tag{9}$$

Example 5

In this example, we consider Hölder Table Function (HTF) because it has many local minima, with four global minima. We have evaluated the function using the input domain of $x_i \in$ [-10, 10]. It is worth mentioning that the HTF is not convex, multimodal and defined in 2-dimensional space. HTF is shown in equation 10:

$$\min -|sin(x)cos(y)exp(|1 - \sqrt{(x^2 + y^2)/\pi}|)| \tag{10}$$



(a) Initial population (b) Generation 2 (c) Generation 4 (d) Generation 4 and optimal solution for HTF

Fig. 9 Process of finding optimal solution by VEA- Example 5

We have applied our proposed VEA to solve HTF. The process of the algorithm, initial population, optimal solution of generations and constraints of the problems have been shown for two iterations in Fig. 9.

Example 6

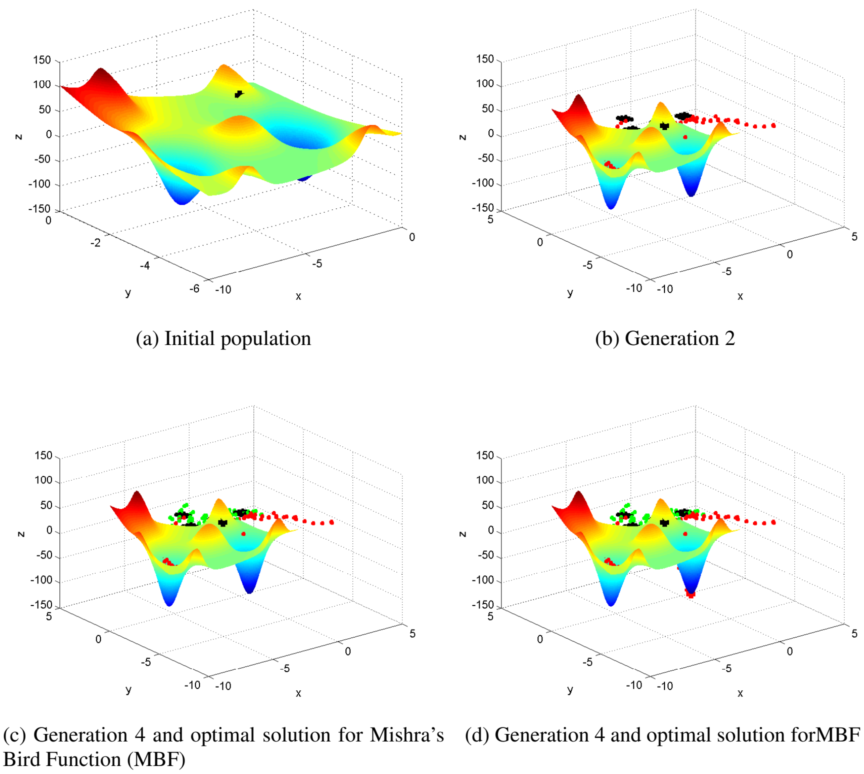
In this example, we consider Mishra's Bird Function (MBF), which is shown in equation 11. The problem has been solved by VEA, the process of the algorithm, initial population, optimal solution of generations and constraints of the problems have been shown for two iterations in Fig. 10.

$$\min sin(x)exp((1-cos(y))^2) + cos(y)exp((1-sin(x))^2) + (x-y)^2 \tag{11}$$

Further, more benchmark examples are required to test and evaluate the proposed VEA. Accordingly, we consider functions such as unimodal, multimodal, fixed-dimension and multimodal. Table 3 shows the examples from 7 to 10 with equations and details. Table 4 and Fig. 11 show the results of VEA for examples 7 to 10, where optimal solutions are found in 1 to 3 iterations.

5.2 VEA Solving Large Size Practical Problems

To show efficiency of the algorithm for real life/size problems, in this section three kinds of practical problems have been solved: large size real linear programming problems and IoV



(a) Initial population

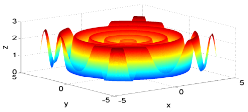
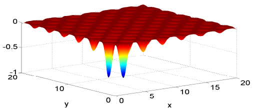
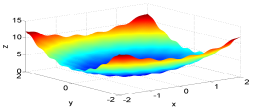
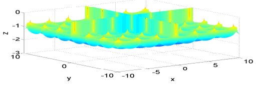
(b) Generation 2

(c) Generation 4 and optimal solution for Mishra's Bird Function (MBF)

(d) Generation 4 and optimal solution forMBF

Fig. 10 Process of finding optimal solution by MV- Example 6

Table 3 Details of Examples 7-10

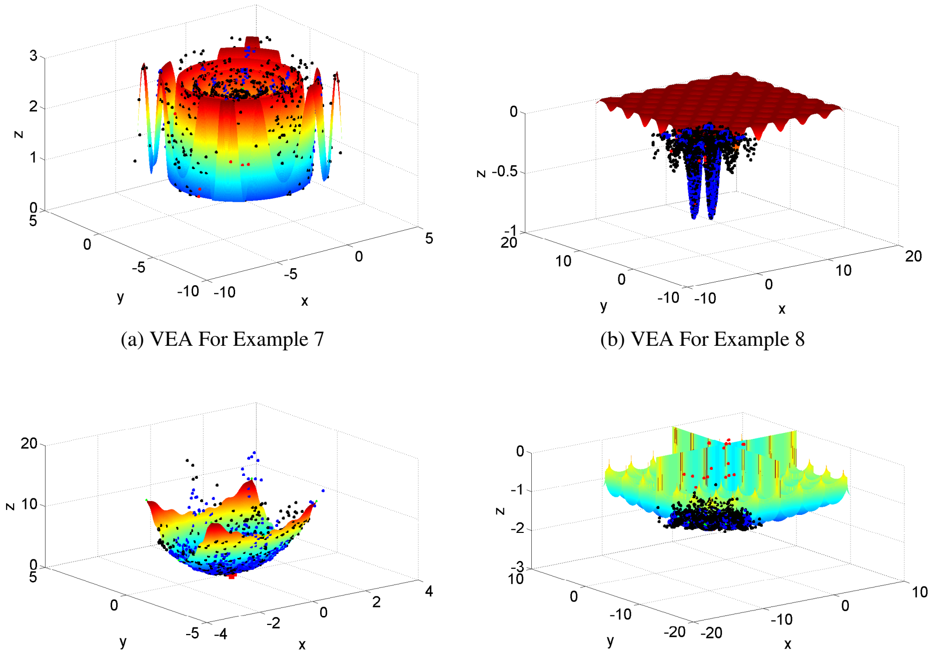
Examples	**Equations**	**Figures**
Example 7 - Salomon Function	$f(\mathbf{x}) = f(x_1, ..., x_n) = 1 - cos(2\pi \sqrt{\sum_{i=1}^{D} x_i^2}) + 0.1 \sqrt{\sum_{i=1}^{D} x_i^2}$	
Example 8 - Keane Function	$f(x,y) = -\frac{\sin^2(x-y)\sin^2(x+y)}{\sqrt{x^2+y^2}}$	
Example 9 - Bohachevsky N. 2 Function	$f(x,y) = x^2 + 2y^2 - 0.3cos(3\pi x)cos(4\pi y) + 0.3$	
Example 10 - Cross-in-Tray Function	$f(x,y) = -0.0001(\|sin(x)sin(y)exp(\|100 - \frac{\sqrt{x^2+y^2}}{\pi}\|)\| + 1)^{0.1}$	

problems. In Matlab, we used ”Linprog” as an exact method based on simplex to solve linear programming problems. Table 5 shows the superiority of VEA in solving large size problems as compared with several Benchmark linear programming functions. Further, absolute error of “Linprog” and VEA in terms of the optimal solution, in Table 6, indicates that the classic method is impractical and inefficient as compared to the proposed VEA. Moreover, finding a suitable feasible solution of transportation problem is very significant, thus VEA was applied to some random transportation problems [3]. Table 7, shows the results of applying VEA in Intelligent transportation problems.

In Table 8, shows the comparison of our proposed VEA with Vogel algorithm, which is the best algorithm in finding feasible solutions of transportation problem. As can be seen, we have shown the superiority of the proposed VEA.

Table 4 Results of VEA for Example 7-10

Examples	**N. Agents**	**N. Iterations**	**Optimal Solution**	**F Min**	ϵ	**Initial Soltion**
Example 7	24	1	(0,0)	0	1	(-2,-3)
Example 8	24	3	(1.39,0)	0.67	1	(5,5)
Example 9	24	1	(0,0)	0	1	(-1,-1)
Example 10	24	2	(1.34,1.34)	-2.06	1	(0,0)



(a) VEA For Example 7 (b) VEA For Example 8

(c) VEA For Example 9 (d) VEA For Example 10

Fig. 11 Process of finding optimal solution by VEA- Examples 7-10

5.3 VEA Solving Route Optimization in IoV Scenario

The objective of this problem is to maximize the connectivity probability and link quality of the available routes from source to destination as illustrated in Fig. 12 [19,43]. The probability of connectivity can be found by real-time estimation of traffic density from source to destination [46]. Further, the maximization process is subject to Signal to Interference and Noise Ratio threshold ($SINR_th$) in order to find more reliable and connected route in urban SDN based vehicular scenarios. The city road networks in vehicular scenario is represented as graph model G(i,e) where i is an intersection and e is the road segment between two intersections [44,45]. Therefore, each optimal route ζ consists of a set of intersections $(i_1, i_2, i_3, i_4, i_5, i_6, ..., i_m)$ and a set of streets $(e_1, e_2, e_3, e_4, e_5, e_6, ..., e_n)$, where $n = m - 1$. According to the aforementioned assumptions, the objective function of the optimization problem can be written as:

Table 5 Results of VEA for more test problems

Name	**Size**	**Optimal**	**Linprog**	**VEA**	**N. Iterations**
agg	489 163	-3.5991767287E+07	-3.9217e+16	- 3.59917e+07	10
qap8	913 1632	2.0350000000E+02	-1.6987e+16	2.144e+02	20
SC50A	51 48	-6.4575077059E+01	-6.5313e+20	-6.4879e+01	5
AFIRO	28 32	-4.6475314286E+02	-1.4505e+29	-4.7361e+02	5
Random Problem	1000 5000	NaN	-400.6831e+36	-124.3891e+07	500

Table 6 Comparison Errors of VEA and Classic Methods

Name	Error of Linprog	Error of VEA
agg	3.9217e+16	67
qap8	1.6987e+16	11
SC50A	6.5313e+20	0.3
AFIRO	1.4505e+29	8.9

Table 7 Comparison among VEA and other algorithms for large size problems

Problems	Size	North-West	Vogel	VEA
Transportation 1	80 20	132804	30123	22150
Transportation 2	100 25	177666	26462	24367
Transportation 3	160 40	185366	85456	62859
Transportation 4	200 50	297629	26566	21578
Transportation 5	210 70	322356	27619	23160
Transportation 6	261 87	245311	152930	119526
Transportation 7	10000 10000	12736903	10321697	5896123

$$\max_{\zeta} \quad F(\zeta) = \lambda_1 \times PC(\zeta) + \lambda_2 \times SINR(\zeta) \tag{12}$$

$$\text{where} \quad PC(\zeta) = \prod_{i=1}^{n} PC(e_i), \qquad SINR(\zeta) = \frac{\sum_{i=1}^{n} SINR(e_i) - \sum_{i=1}^{n} SINR_{th}(e_i))}{\sum_{i=1}^{n} SINR(e_i)}, \tag{13}$$

$$\text{subject to} \quad SINR(\zeta) \geq SINR_{th}(\zeta). \tag{14}$$

where F(ζ) is defined as the objective function with a set of routes ζ from source to destination. $\lambda 1$ and $\lambda 2$ are the weights that empirically set in the simulation and their summation is equal to 1. PC(ζ) and SINR(ζ) connectivity and reliability of routes respectively. $PC(\zeta)$ and $SINR(\zeta)$ connectivity and reliability of routes respectively. $PC(e_i)$ and $SINR(e_i)$ representing the street's connectivity and link reliability. Fig. 12 illustrates the routing process in SDIoV [19].

This problem is addressed by both mathematical and heuristic algorithms. Laying Chicken Algorithm (LCA) [12] has been used to find optimal route from source to destination [19]. The comparison of results of LCA and VEA are provided in Table 9.

Table 8 Improvement amount of Vogel algorithm by VEA

Problems	Size	Vogel	VEA	Improvement byVEA
Transportation 1	80 20	30123	22150	0.26
Transportation 2	100 25	26462	24367	0.08
Transportation 3	160 40	85456	62859	0.26
Transportation 4	200 50	26566	21578	0.19
Transportation 5	210 70	27619	23160	0.16
Transportation 6	261 87	152930	119526	0.22

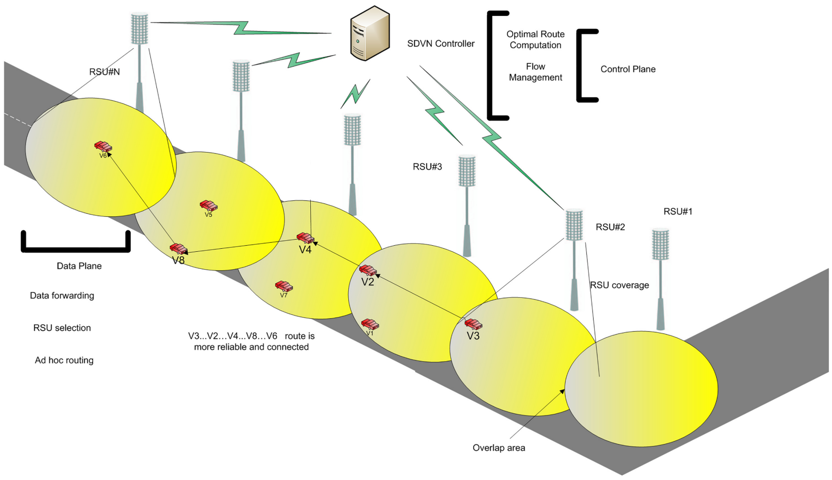


Fig. 12 Optimal routing process in IoV environment

Table 9 Comparison of LCA and VEA for internet of vehicles

Problems	Size	Best Solution by LCA	Best Solution by VEA
IoV 1	100 100	775.8550	917.3405
IoV 2	200 200	9.9319e+03	1.3014e+04
IoV 3	500 500	5.8147e+04	6.9372e+04
IoV 4	1000 1000	2.5991e+05	2.8461e+05
IoV 5	2000 2000	9.8622e+05	1.2831e+06
IoV 6	5000 5000	6.2266e+06	6.6281e+06
IoV 7	10000 10000	2.4950e+07	2.7145e+07
IoV 8	30000 30000	3.7632e+09	3.7916e+09

Table 10 Improvement of VEA and LCA from their Random Initial Solutions (RIS) in five iterations

Problems	Size	Improvement of RIS by LCA	Improvement of RIS by VEA
IoV 1	100 100	0.031	0.221
IoV 2	200 200	0.005	0.318
IoV 3	500 500	0.008	0.202
IoV 4	1000 1000	0.002	0.097
IoV 5	2000 2000	0.002	0.303
IoV 6	5000 5000	0.001	0.064
IoV 7	10000 10000	0.0001	0.081
IoV 8	30000 30000	0.0001	0.069

For each problem an initial solution has been generated randomly and these initial solution are different for both LCA and VEA algorithms. Table 10 shows improvement of their initial solutions after five iterations.

5.4 Comparison of VEA with Other Optimization Techniques

In this section, VEA is compared with other techniques, VEA is used to solve two different categories of test functions: unimodal and multi-modal. Unimodal test functions have just a global optimum but multi-modal test functions have a global optimum as well as multiple local optima. Details of these benchmark functions have been shown in Table 11. For the verification of the results, proposed algorithm is compared with Multi-Verse Optimizer (MVO) [26], Grey Wolf Optimizer (GWO) [27], PSO and GA.

Note that the number of agents is set on 50 and the maximum number of iterations is equal to 100 and epsilon is 0.1, also the algorithm is run 50 times. The results presented in Table 12 shows that the proposed algorithm is able to provide very competitive and efficient performance on both the unimodal and multi-modal test functions. Ave. and Std. are average results and corresponding standard deviations respectively. Low standard deviation of VEA is significant, which indicates that the values tend to be close to the mean of the set.

6 Conclusion

A novel meta-heuristic optimization algorithm has been proposed in this paper, which is inspired from a natural event of volcano eruption. The proposed algorithm has formulated a new optimizer for solving numerous sorts of continuous and discrete optimization problems. Utilizing the natural process of volcano eruption, our proposed Volcano Eruption Algorithm (VEA) has been formulated and been used as an optimizer. The significance of our proposed VEA lied in its robustness in producing a wide-range set of the initial population, movement pattern across the solution's space, explosion and eruption in the space. This was achieved from the concept of the volcano eruption's nature, which has contributed significantly in improving the optimization process. Therefore, the proposed VEA optimizer could achieve an acceptable computational complexity with noticeable improved performance in comparison with the state of the art. Numerical results presented in this paper have shown that our proposed VEA could significantly contribute in solving wide-range of linear, non-linear, multi-level, multi-objective, and transportation based on IoV complex optimization problems. The following briefly outline some areas for future work to be further studied:

1. Explore the possibility of solving some NP hard problems such as travelling salesman problem using the proposed VEA.
2. VEA should be attempted in solving problems that involving big data as it has appropriate complexity.
3. The algorithm should be extended for solving discrete problems such as shortest path problem, etc.
4. Combination of proposed algorithm as an inspired approach with exact methods. For example finding an approximate gradient vector by VEA for using methods such as simplex, which uses gradient directly.
5. Implementation of such similar ideas like floods, hurricanes, earthquakes and others.

References

1. J. Kennedy and R. C. Eberhart. *Particle Swarm Optimization. In Proceedings of IEEE.* International Conference on Neural Networks, Perth, Australia, pp. 1942–1948, (1995).
2. Eghbal Hosseini. *Presentation and Solving Non-Linear Quad-Level Programming Problem Utilizing a Heuristic Approach Based on Taylor Theorem.*

Table 11 Optimization Test Functions

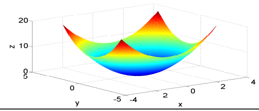
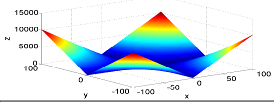
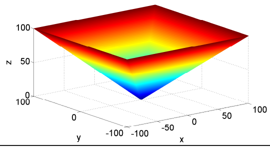
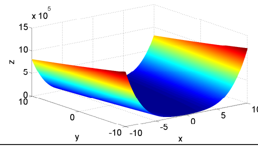
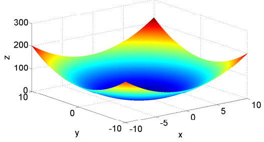
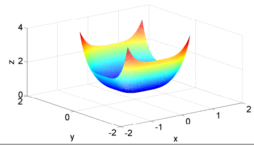
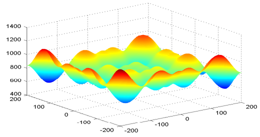
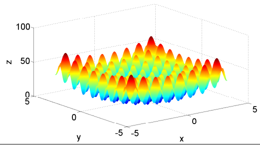
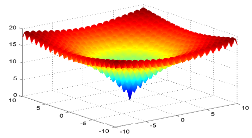
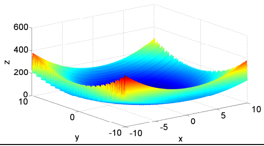
Functions	Equations	Figures
F1-Sphere Function	$f(\mathbf{x}) = f(x_1, ..., x_n) = \max_{i=1,...,n} \lvert x_i \rvert$	
F2-Schwefel 2.22 Function	$f(\mathbf{x}) = f(x_1, ..., x_n) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	
F3-Schwefel 2.21 Function	$f(\mathbf{x}) = f(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} x_i^2$	
F4-Rosenbrock Function	$f(\mathbf{x}) = f(x_1, ..., x_n) = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} x_i + n/4)$	
F5-Zakharov Function	$f(\mathbf{x}) = f(x_1, ..., x_n) = -a.exp(-b\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - exp(\frac{1}{n}\sum_{i=1}^{n} cos(cx_i)) + a + exp(1)$	
F6-Quartic Function	$f(x, y) = sin^2(3\pi x) + (x-1)^2(1 + sin^2(3\pi y)) + (y-1)^2(1 + sin^2(2\pi y))$	
F7-Schwefel Function	$f(\mathbf{x}) = f(x_1, ..., x_n) = \sum_{i=1}^{n} ix_i^4 + \text{random}[0, 1)$	
F8-Rastrigin Function	$f(x, y) = 10n + \sum_{i=1}^{n} (x_i^2 - 10cos(2\pi x_i))$	
F9-Ackley Function	$f(x, y) = \sum_{i=1}^{n} [b(x_{i+1} - x_i^2)^2 + (a - x_i)^2]$	
F10-Levi N. 13 Function	$f(\mathbf{x}) = f(x_1, x_2, ..., x_n) = 418.9829d - \sum_{i=1}^{n} x_i sin(\sqrt{\lvert x_i \rvert})$	

Table 12 Comparison of VEA and other methods with benchmarks

Functions	VEA		MVO		GWO		PSO		GA	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
F1	1.7367	0.5892	2.08583	0.648651	2319.19	1237.109	3.552364	2.85373	27,187.58	2745.82
F2	5.1245	3.2689	15.92479	44.7459	14.43166	5.923015	8.716272	4.929157	68.6618	6.062311
F3	1.1678	1.2689	3.123005	1.582907	13.09729	11.3469	21.5169	6.71628	62.99326	2.535643
F4	945.148	813.695	1272.13	1479.477	3425,462	3304,309	1132.486	1357.967	65,361,620	29,714,021
F5	1.1834	0.5277	2.29495	0.630813	5009.442	3028.875	86.62074	147.3067	49,574.1	8545.149
F6	0.01423	0.01125	0.051991	0.029606	0.408082	0.119544	0.577434	0.318544	18.72524	4.935256
F7	-8845.38	567.38	-11,720	937.1975	-10,739	1162.793	-6727	1352.882	-10,698	602.3045
F8	12.4784	8.3213	118.046	39.34364	89.13475	37.95765	99.83202	24.62872	273.2519	29.55218
F9	1.3269	0.8263	4.074904	5.501546	9.452571	3.467608	4.295044	1.308386	18.59657	0.351737
F10	0.6193	0.1014	2.459953	0.791886	3,200,008	6,746,208	13.38384	8.969122	2.21e+08	1.1e+08

3. Eghbal Hosseini. *Solving Linear Tri-level Programming Problem Using Heuristic Method Based on Bisection Algorithm*. Asian Journal of Scientific Research 10 (4) (2017) .
4. Eghbal Hosseini. *Three new methods to find initial basic feasible solution of transportation problems*. Applied Mathematical Sciences 11 (no. 37,), 1803-1814. (2017).
5. D. Karaboga and B. Basturk. *A Powerful And Efficient Algorithm For Numerical Function Optimization: Artificial Bee Colony (ABC)*. Journal of Global Optimization,Springer Netherlands, Vol. 39, No. 3, pp: 459-471, (2007).
6. Gandomi, A.H., Alavi, A.H. *Krill herd: A new bio-inspired optimization algorithm*. Communications in Nonlinear Science and Numerical Simulation 17, 4831–4845 (2012).
7. Yang, X.S. *Bat algorithm: literature review and applications*. International Journal of Bio-inspired Computation 5(3), 141–149 (2013).
8. Cuevas, E., Cienfuegos, M., Zaldivar, D., Cisneros, M. *A swarm optimization algorithm inspired in the behavior of the social-spider*. Expert Systems with Applications 40, 6374–6384 (2013).
9. Xianbing Meng, Yu Liu, Xiaozhi Gao, and Hengzhen Zhang. *A New Bio-inspired Algorithm: Chicken Swarm Optimization*. ICSI 2014, Part I, LNCS 8794, pp. 86–94 (2014).
10. Xin She Yang. *Nature- Inspired Meta-Heuristic Algorithms*. University of Cambridge 2010.
11. Eghbal Hosseini. *Big Bang Algorithm: A New Meta-heuristic Approach for Solving Optimization Problems*. Asian Journal of Applied Sciences 10 (4), 334-344 (2017).
12. Eghbal Hosseini. *Laying Chicken Algorithm: A New Meta-Heuristic Approach to Solve Continuous Programming Problems*. Journal of Applied and Computational Mathematics 6 (1) (2017).
13. Eghbal Hosseini, IN Kamalabadi. *A Genetic Approach for Solving Bi-Level Programming Problems*. Advanced Modeling and Optimization 15 (3) (2013).
14. Eghbal Hosseini, IN Kamalabadi. *Line search and genetic approaches for solving linear tri-level programming problem*. International Journal of Management, Accounting and Economics 1 (4) (2015).
15. JT Salzer, WA Thelen, MR James, TR Walter, S Moran, R Denlinger. *Volcano dome dynamics at Mount St. Helens: Deformation and intermittent subsidence monitored by seismicity and camera imagery pixel offsets*. Journal of Geophysical Research: Solid Earth 121 (11), 7882-7902 (2016).
16. M. Bazzara. *non-linear programming Theory and Algorithms*. Wiley, Inc, New York (2007).
17. Eghbal Hosseini. , Isa Nakhai Kamalabadi *Solving Linear Bi-level Programming Problem Using Two New Approaches Based on Line Search and Taylor Methods*. Management sciences and Education, Vol.2 (6), 243-252 November (2014).
18. M. Bazzara. *linear programming and Network Flows*. Wiley, Inc, New York (2010).
19. Ghafoor Kayhan; Kong Linghe ; Danda Rawat; Hosseini Eghbal ; Sadiq Ali. *Quality of Service Aware Routing Protocol in Software-Defined Internet of Vehicles*. IEEE Internet of Things Journal, October (2018).
20. K. Deb;L. Thiele; M. Laumanns; E. Zitzleri. *Scalable multi-objective optimization test problems*. in Proc. IEEE Congr. Evol. Comput., Honolulu, HI, USA, pp. 825–830(2002).
21. Chugh, Tinkle; Jin, Yaochu; Miettinen, Kaisa; Hakanen, Jussi; Sindhya, Karthik. *A Surrogate-assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-objective Optimization*. IEEE Transactions on Evolutionary Computation, 22 (1), 129-142 (2018).
22. Mirjalili, S; Dong, J. S.; Lewis, A.; Sadiq, A. S. *Particle Swarm Optimization: Theory, Literature Review, and Application in Airfoil Design*. In Nature-Inspired Optimizers, (pp. 167-184), Springer, Cham (2020).

23. Mirjalili, S; Lewis, A.; Sadiq, A. S. *Autonomous particles groups for particle swarm optimization*. Arabian Journal for Science and Engineering, 39(6), 4683-4697 (2014).
24. Shahjehan, Waleed; Riaz, Asad; Khan, Imran; Sadiq, Ali Safaa; Khan, Suleman; Khan, Muhammad Khurram. *BAT Algorithm based Beamforming for mmWave Massive MIMO Systems*. International Journal of Communication Systems, DOI: 10.1002/dac.4182 (2019).
25. Mafarja, M; Mirjalili, S. *Whale optimization approaches for wrapper feature selection*. Applied Soft Computing, 62, 441-453. (2018).
26. Javidy, B; Hatamlou, A; Mirjalili, S. *Ions motion algorithm for solving optimization problems*. Applied Soft Computing, 32, 72-79 (2015).
27. Mirjalili, S; Mirjalili, S; M.; Lewis, A. *Grey wolf optimizer*. Applied Soft Computing, 69, 46-61 (2014).
28. Mirjalili, S; Gandomi, A. H. *Chaotic gravitational constants for the gravitational search algorithm*. Applied Soft Computing, 53, 407-419 (2017).
29. Sadiq, A. S.; Faris, H.; Ala'M, A. Z.; Mirjalili, S.; Ghafoor, K. Z. *Fraud Detection Model Based on Multi-Verse Features Extraction Approach for Smart City Applications*. In Smart Cities Cybersecurity and Privacy (pp. 241-251). Elsevier (2019).
30. Mirjalili, S.; Dong, J. S.; Sadiq, A. S.; Faris, H. *Genetic Algorithm: Theory, Literature Review, and Application in Image Reconstruction*. In Nature-Inspired Optimizers (pp. 69-85). Springer, Cham (2020).
31. Claudio Gentile, Shuai Li. *Collaborative Filtering Bandits*. the 39th International ACM SIGIR conference (2015).
32. Nathan Korda, Balazs Szorenyi, Shuai Li. *Distributed Clustering of Linear Bandits in Peer to Peer Networks*. Proceedings of the 33 rd International Conference on Machine Learning, New York, NY, USA (2016).
33. Fei Hao, Doo-Soon Park, Shuai Li. *Mining Maximal Cliques from a Fuzzy Graph*. Sustainability, 8(6), 553, (2016).
34. Arikrishna Narasimhan, Shuai Li, Purushottam Kar, Sanjay Chawla, Fabrizio Sebastiani. *Stochastic Optimization Techniques for Quantification Performance Measures*. KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016).
35. Boyd, S., Vandenberghe, L. *Convex optimization*. Cambridge University Press. (2004).
36. Lera, D., Sergeyev, Y. D *GOSH: Derivative-free global optimization using multi-dimensional space-filling curves*. Journal of Global Optimization, 71(1), 193–211 (2018).
37. Faramarzi, A., Afshar, M. H. *A novel hybrid cellular automata–linear programming approach for the optimal sizing of planar truss structures*. Civil Engineering and Environmental Systems, 31(3), 209–228 (2014).
38. Sergeyev, Y. D., Kvasov, D. E., Mukhametzhanov, M. S. *On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget*. Scientific Reports, 8(1), 1–9. (2018).
39. M Dorigo and G DiCaro *The ant colony meta-heuristic*. in new ideas in optimization, (1999).
40. Marco Dorigo, Gianni Di Caro, and Luca M Gambardella *Ant algorithms for discrete optimization*. Artificial life, 5(2):137–172, , (1999).
41. Yang, X. S. *Firefly algorithm*. Nature-inspired metaheuristic algorithms, 20, 79-90, (2008).
42. Erol, O. K., Eksin, I. *A new optimization method: big bang–big crunch*. Advances in Engineering Software, 37(2), 106-111, (2006).
43. Ghafoor, Kayhan Zrar and Guizani, Mohsen and Kong, Linghe and Maghdid, Halgurd S and Jasim, Khalid Fadhil *Enabling Efficient Coexistence of DSRC and C-V2X in Vehicular Networks*. IEEE Wireless Communications, (2019).
44. Ghafoor, Kayhan Zrar and Bakar, Kamalrulnizam Abu and Lloret, Jaime and Khokhar, Rashid Hafeez and Lee, Kevin C *Intelligent beaconless geographical forwarding for urban vehicular environments*. Wireless networks, Springer, 19(3), 345-362 (2013).
45. Sadiq, Ali Safa and Khan, Suleman and Ghafoor, Kayhan Zrar and Guizani, Mohsen and Mirjalili, Seyedali *Transmission power adaption scheme for improving IoV awareness exploiting: evaluation weighted matrix based on piggybacked information*. Computer Networks, Elsevier, 19(3), 147–159 (2018).
46. Kong, Linghe and Xue, Guangtao and Ghafoor, Kayhan Zara and Hussain, Rasheed and Sheng, Hao and Zeng, Peng *Real-Time Density Detection in Connected Vehicles: Design and Implementation*. IEEE Communications Magazine, IEEE, 56(10), 64–70 (2018).